Auxiliary feature learning for small dataset regularization

Spyridon Kavvadias
University of Essex
Colchester, UK
sk18700@essex.ac.uk

ABSTRACT

There has been a rise of methods that aim to learn from small datasets. The goal of this project is to see if the use of an autoencoder to extract useful features can help a neural network learn a discriminative classifier. Three datasets had their dimensionality reduced by an autoencoder. The autoencoder's output was then used to train a discriminative neural network. The accuracy was compared to the same architecture neural network that trained on the unedited dataset. The results were similar, so the conclusion is that the use of an autoencoder does not improve the learning procedure.

INTRODUCTION

The use of an autoencoder in order to reduce the dimensionality of the data before feeding it to a neural network could be vital to enhance the learning process. However, dimensionality cannot be reduced without data loss. In this project, the aim is to decide whether that loss can be detrimental to the accuracy of a discriminative neural network or not. Furthermore, the idea that smaller datasets are also effective for learning is also explored.

BACKGROUND

Autoencoders are simple neural networks that aim to turn inputs into outputs with little distortion. They were first introduced in the 1980s by Hinton, Rumelhart and Williams[1] who tried to deal with the problem of using the input data in a neural network as the way to effectively backpropagate. Autoencoders have recently come to attention again[2][3][4][5], especially "Restricted Boltzmann Machines", as a way to tune the architecture of a system. However, the understanding regarding autoencoders hasn't been greatly enhanced, with few exclusions[6][7].

An autoencoder is an unsupervised training algorithm neural network that applies backpropagation and sets the target values to be equal to the inputs, as seen in figure 1.

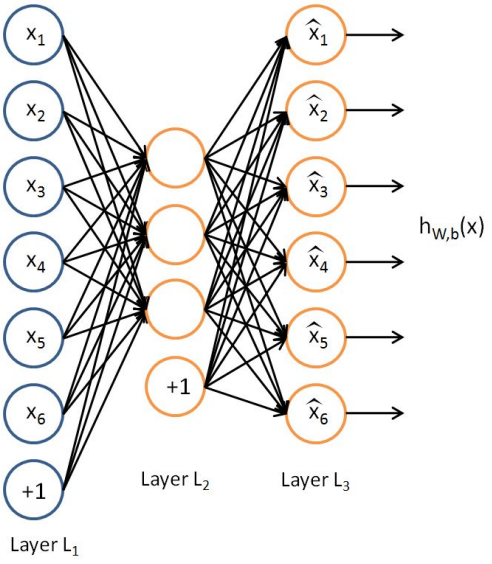

Figure 1: Simple autoencoder

An autoencoder tries to approximate the input to the output, so that they are similar. By having the hidden layer nodes less than the input nodes, the network is forced to learn a compressed representation of the input, which it must reconstruct at the output nodes. It is a way to reduce the dimensionality of the data in order to also reduce the complexity. In this project, an autoencoder is created for every dataset used, then the compressed representation is used to train a neural network.

An artificial neural network (ANN) is inspired by the way the human brain works. It consists of connected nodes, called artificial neurons. An artificial neuron receives a signal, processes it and signals other neurons that are connected to it. The signal mentioned is a real number, and the output is calculated by a non-linear function of the sum of its inputs. Neurons are connected by the edges. The edges have a weight that gets adjusted during the training phase.

The training phase is when the neural network is trained to recognise certain patterns in a dataset. In this project, there is a classification variable in each dataset, which is effectively its output. After each epoch, the dataset is validated on a separate set of data and the validation accuracy is calculated. An example of a simple neural network is seen in figure 2.

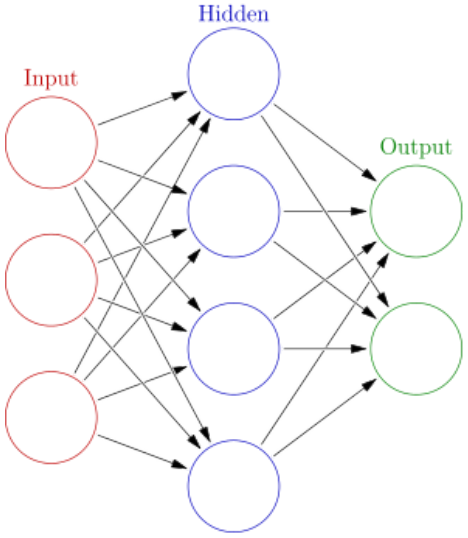

Figure 2: Simple neural network

METHODOLOGY

The datasets used in this project were the following;

1. Breast Cancer Coimbra Data Set[1]. 10 predictors are used, all quantitative, along with a binary dependent variable, indicating the presence or absence of breast cancer. The predictors are anthropometric data and parameters which can be gathered in routine blood analysis. A prediction model based on these predictors can be used as a biomarker of breast cancer. The clinical features were observed or measured for 64 patients with breast cancer and 52 healthy controls.

2. Dermatology Data Set[2]. This dataset contains 34 attributes, 33 of which are linear valued and one of them is nominal. The aim for this dataset is to determine the type of Erythemato-Squamous Disease. The diseases in this group are psoriasis(1), seboreic dermatitis(2), lichen planus(3), pityasis rosea(4), cronic dermatitis(5) and pityriasis rubra pilaris(6). Patients were first evaluated clinically with 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. In this dataset, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1,2 indicate the relative intermediate values.

3. Audit Data Set[3]. An exhaustive one year non-confidential data in the year 2015 to 2016 of firms is collected from the Auditor Office of India to build a predictor for classifying suspicious firms. The classification model should be able to predict the fraudulent firm on the basis of present and historical risk factors.

The autoencoder structure is defined. In each dataset the dimensionality has been reduced to half in the hidden layer of the autoencoder. In figure 3, the breast cancer dataset, which has 9 inputs excluding the classification variable has been reduced to 5. The dermatology dataset has been reduced from 34 inputs to 17, while the Audit dataset has been reduced from 25 inputs to 12.

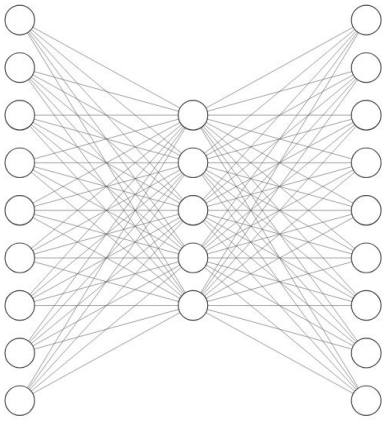

Figure 3: Breast cancer dataset autoencoder

All of the autoencoders are trained for 500 epochs and the hidden layer is saved as a separate model in the "weights" folder inside the project. These models are then used to create the reduced dataset which is then used to train a neural network.

Input:[[0.61538464,0.79963964,0.24822696,0.0356072,0.02318621,0.19659062,0.10462296,0.04426264,0.125019]] Encoded:[[0.32710564,0,0.4705032,0,0]] Decoded:[[0.52017355,0.734957,0.36580807,0.12874317,0.06099538,0.33554542,0.2049478,0.29385874,0.22156872]]

Table 1: Reduced breast cancer dataset entry

An example of a reduced entry of the breast cancer dataset can be seen in table 1. The input is 9 entries and the encoded output is just 5. The loss of this specific autoencoder is 0.0308, however the loss in the decoded data can already be seen.

After the autoencoders are trained, the next step is to train a neural network based on the output of their hidden layer, or the reduced encoded data. In figure 4 the neural network to train on the breast cancer dataset is shown.

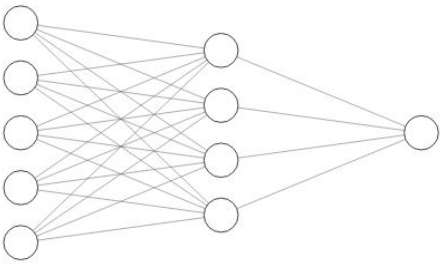

Figure 4: Breast cancer dataset neural network

The hidden layer uses the 'relu' activation function, while the output uses 'sigmoid'. The model is trained using the 'SGD'(Stochastic gradient descent) optimizer, which allows for the tuning of the learning rate and momentum variables. They have been tuned at 0.4 and 0.001 respectively. The validation accuracy for this model fluctuates around 75%.

The dermatology dataset's neural network consists of an input of 17 nodes, reduced by the autoencoder. The hidden layer consists of 16 nodes and the output is 1 node. The hidden layer uses the 'relu' activation function, while the output uses 'sigmoid. The stochastic gradient descent optimizer is used and the learning rate and momentum are tuned to 0.01 and 0.4 respectively. The validation accuracy is very bad, although the best it can get. It ranges from 25% to 40%. Figure 5 shows the architecture of the neural network.

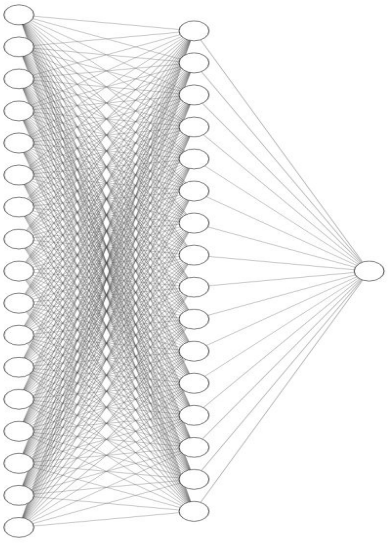

Figure 5: Dermatology dataset neural network

The neural network for the audit dataset consists of 12 input nodes, 11 first hidden layer nodes, 24 second hidden layer nodes

and 11 third hidden layer nodes, the output following. The stochastic gradient descent optimizer is used and the learning rate and momentum parameters are tuned at 0.01 and 0 respectively. The validation accuracy is very good, more than 90%, with the highest at 97%. Figure 6 shows the architecture.

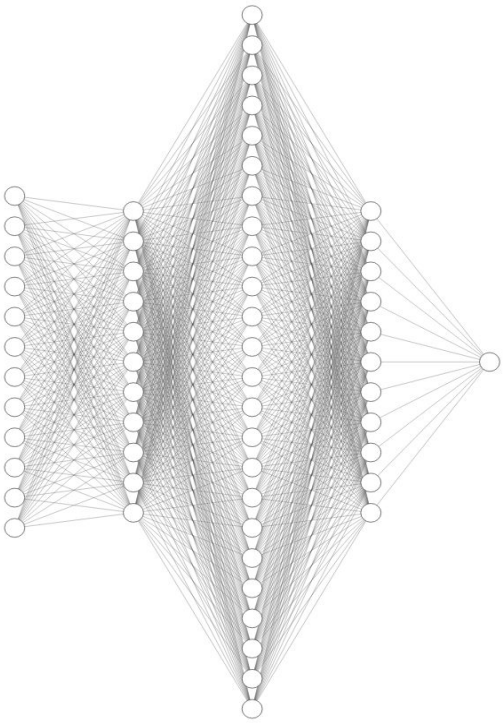

Figure 6: Audit dataset neural network

EXPERIMENTS

The first step is to clean the data. The initial line which contained the column names was cleared from each data set. Following that, in the dermatology dataset, there were some rows which had missing values for the age column, specified by a question mark. Instead of deleting the whole "Age" column, these specific rows were deleted from the dataset as they couldn't be of any value. Moving on to the audit dataset, there were rows which had text instead of a number in the "LOCATION_ID" column (rows #353, #357, #369). They were deleted as they could be of no value. Furthermore, there were some entries which contained blank values. The entire rows were deleted since they couldn't be used to train the neural network.

To continue, the datasets need to be checked so that they are not imbalanced, meaning that the are equal occurences of the classification variables.

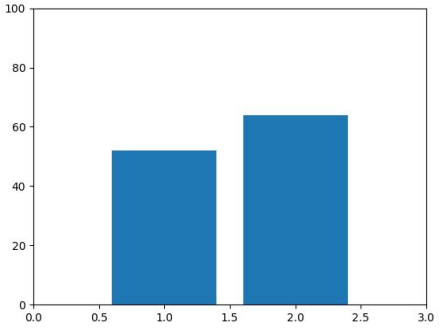

Figure 7: Breast cancer dataset bar chart

In the breast cancer dataset, the classification variable can take two values, either 1 or 2. In figure 7 it is clear that the dataset is almost equally split so there is no need to perform any actions.

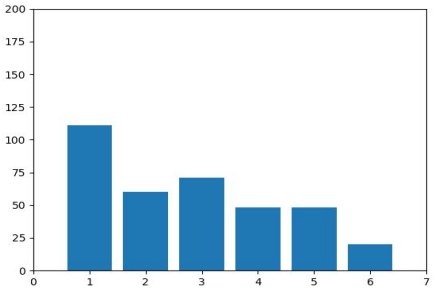

Figure 8: Initial dermatology dataset bar chart

Regarding the dermatology dataset, the classification variable can take values from 1-6. As shown in figure 8, their distribution is not equal, so that would mean the neural network will not be trained to recognize all cases equally. In order to fix this dataset, some entries which have a "1" outcome need to be removed. There is also a need to increase the entries with a "6" outcome. This dataset needs to have a combination of undersampling and oversampling. The only problem is that there are only 20 instances with a "6" outcome, so they were duplicated twice in order to be oversampled. The final outcome is in figure 9.

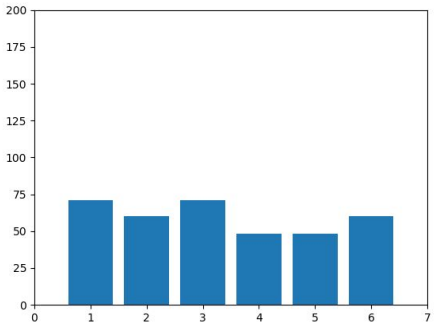

Figure 9: Dermatology dataset

Figure 10 shows the distribution for the Audit dataset. The outcome is either 1 or 0, which determines the risk. It is in perfect balance, so there is no need to edit it.

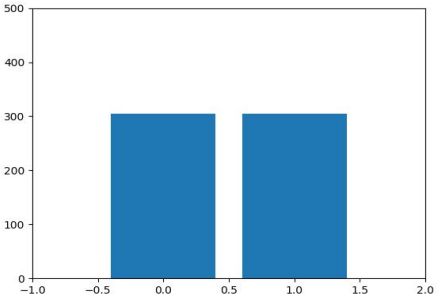

Figure 10: Audit dataset

Another important concept is the data split equality. During the splitting of the data into training and testing data, the ratio between the classification variable values has to remain the same as the original dataset before splitting. This can help ensure that the neural network will train on all cases equally and will not only learn a few of the cases. This has been ensured during the process, by continuously splitting the data until the wanted ratio is achieved on all datasets.

The most important task was to tune the parameters of the encoders and the neural networks. Since there is no formula, this has to be done through trial and error. The parameters used are the optimal ones for the current architecture.

DISCUSSION

One additional important factor is to compare the results using different percentages of the datasets in order to find out which is more effective, and also if using the autoencoder is more effective in datasets without many entries. In tables 2,3 and 4 below, the results of the tests can be seen. The tables show the impact the size of the dataset has on the final results. The final results are splitted into the neural network trained using the output of the autoencoder(AE), and training the neural network(NN) without the autoencoder.

Data %	Breast cancer dataset		
	AE loss (%)	AE and NN accuracy(%)	NN only accuracy (%)
30%	3.136	66	68.4
50%	3.51	63	78.2
70%	3.39	60	72.8
100%	3.1	63.2	76.6

Table 2: Breast cancer dataset test run

Data %	Dermatology dataset		
	AE loss (%)	AE and NN accuracy(%)	NN only accuracy (%)
30%	7.4	21	33.6
50%	6.1	32.4	36.2
70%	5.9	29.6	34.2
100%	5.7	32.4	35.6

Table 3: Dermatology dataset test run

Data %	Audit dataset		
	AE loss (%)	AE and NN accuracy(%)	NN only accuracy (%)
30%	5.2	80.2	95.4
50%	3.7	84.8	96.4
70%	3.4	91.2	96.2
100%	2.9	93	95.8

Table 4: Audit dataset test run

The results were acquired through changing the percentage of the dataset that is used in the training of the autoencoder and the neural network. Each dataset was tested 10 times for each data percentage and the average value of the results was saved.

It is clear that just using a neural network is better in all cases. The size of the dataset does not seem to have any effect on aiding the autoencoder reduced dataset perform better. Although, in all cases, reducing the dataset size to half does not have a bad impact on the final accuracy, either with or without the autoencoder.

CONCLUSION

It is clear, based on the results, that using an autoencoder to reduce the dimension of data before using it to train a neural network does not make a difference in the final accuracy. However, since the combinations between the possible neural network architectures and the parameters are endless, there may be a more optimal solution which proves the opposite. The results are based on the architecture and the parameters that were chosen to be the most optimal through trial.

REFERENCES

1. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986.
2. G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504, 2006.
3. G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504, 2006.
4. Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, Large-Scale Kernel Machines. MIT Press, 2007.
5. Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11:625–660, February 2010.
6. I. Sutskever and G.E. Hinton. Deep, narrow sigmoid belief networks are universal approximators. Neural Computation, 20(11):2629–2636, 2008.
7. G. Montufar and N. Ay. Refinements of Universal Approximation Results for Deep Belief Networks and Restricted Boltzmann Machines. Neural Computation, pages 1–14, 2011. ISSN 0899-7667.